

# Delta highlighting

```
foldl op:fun sumSquares x y → 416
  + * x
    x
  * y
    y
  z:0
  xs:Cons(4,Cons(2,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 20
      z:z
      xs:xs'
    → case xs of
      Nil → z
      Cons(x,xs') →
        op x
        foldl op 4
          z
          xs'
        ...
    ...
  → + * x16
    x
  * y400
    y
```

# Delta highlighting

```
foldl op:fun sumSquares x y → 416
+ * x
  x
  * y
  y
z:0
xs:Cons(4,Cons(2,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 20
      z:z
      xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x
          foldl op 4
            z
            xs'
            ...
          ...
      → + * x 16
        x
        * y 400
        y
```

edits highlighted

# Delta highlighting

```
foldl op:fun sumSquares x y → 436
+ * x
  x
  * y
  y

z:0
xs:Cons(6,Cons(2,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 20
      z:z
      xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x
          foldl op 4
            z
            xs'
            ...
          ...
    → + * x 36
      x
      * y 400
      y
```

edits highlighted

# Delta highlighting

```
foldl op:fun sumSquares x y →  
  + * x  
    x  
  * y  
    y  
z:0  
xs:Cons(6,Cons(2,Cons(2,Nil)))  
→ case xs of  
  Nil → z  
  Cons(x,xs') →  
    op x:x  
    y:foldl op:op 20  
      z:z  
      xs:xs'  
    → case xs of  
      Nil → z  
      Cons(x,xs') →  
        op x  
        foldl op 4  
          z  
          xs'  
        ...  
    → + * x 36  
      x  
    * y 400  
      y
```

and consequences of edits

436

36

400

# Delta highlighting

```
foldl op:fun sumSquares x y → 436
  + * x
    x
  * y
    y
  z:0
  xs:Cons(6,Cons(2,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 20
      z:z
      xs:xs'
    → case xs of
      Nil → z
      Cons(x,xs') →
        op x
        foldl op 4
          z
          xs'
        ...
    ...
  → + * x 36
    x
  * y 400
    y
```

# Delta highlighting

```
foldl op:fun sumSquares x y → 436
  + * x
    x
  * y
    y

z:0
xs:Cons(6,Cons(2,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 20
      z:z
      xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x
          foldl op 4
            z
            xs'
            ...
          ...
      → + * x 36
        x
        * y 400
        y
```

another edit

# Delta highlighting

```
foldl op:fun sumSquares x y → 661
  + * x
    x
  * y
    y

z:0
xs:Cons(6,Cons(3,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 25
      z:z
      xs:xs'
    → case xs of
      Nil → z
      Cons(x,xs') →
        op x
        foldl op 4
          z
          xs'
        ...
    ...
  → + * x 36
    x
  * y 625
    y
```

another edit

# Delta highlighting

```
foldl op:fun sumSquares x y → 661
  + * x
    x
  * y
    y
z:0
xs:Cons(6,Cons(3,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 25
      z:z
      xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x
          foldl op 4
            z
            xs'
          ...
          ...
      → + * x 36
        x
        * y 625
        y
```

some consequences hidden under “...”



# Delta highlighting

```
foldl op:fun sumSquares x y → 661
  + * x
    x
  * y
    y
z:0
xs:Cons(6,Cons(3,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 25
      z:z
      xs:xs'
    → case xs of
      Nil → z
      Cons(x,xs') →
        op x:x
        y:foldl op 4
          z
          xs'
          ...
        → + * x9
          x
          * y16
          y
    → + * x36
      x
      * y625
      y
```

some consequences hidden under “...”

# Structural deltas

```
foldl op:fun sumSquares x y → 661
  + * x
    x
  * y
    y
  z:0
  xs:Cons(6,Cons(3,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 25
      z:z
      xs:xs'
    → case xs of
      Nil → z
      Cons(x,xs') →
        op x:x
        y:foldl op 4
          z
          xs'
        ...
        → + * x9
            x
          * y16
            y
        → + * x36
            x
          * y625
            y
```

# Structural editing

```
foldl op:fun sumSquares x y → 661
  + * x
    x
  * y
    y
z:0
xs:Cons(6,Cons(3,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
      y:foldl op:op 25
        z:z
        xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x:x
            y:foldl op 4
              z
              xs
            ...
          → + * x9
              x
              * y16
              y
            → + * x36
                x
                * y625
                y
```

edit a function whilst  
browsing its execution

# Structural editing

```
foldl op:fun sumSquares x y → 36
  / + * x
    x
    * y
    y
  2

z:0
xs:Cons(6,Cons(3,Cons(2,Nil)))

→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 6
      z:z
      xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x:x
          y:foldl op 2
            z
            xs'
            ...
            → / + * x913
                x
                * y4
                y
              2
            → / + * x3672
                x
                * y36
                y
              2
```

new nodes are green

# Structural editing

```
foldl op:fun sumSquares x y → 36
  / + * x
    x
    * y
    y
  2

z:0
xs:Cons(6,Cons(3,Cons(2,Nil)))

→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op 6
      z:z
      xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x:x
          y:foldl op 2
            z
            xs'
            ...
            → / + * x913
              x
              * y4
              y
              2

→ / + * x3672
  x
  * y36
  y
  2
```

consequences reflected immediately in execution

# Structural editing

# delta-driven programming

```
foldl op:fun sumSquares x y →
  / + * x
  | x
  | * y
  | y
  2
z:0
xs:Cons(6,Cons(3,Cons(2,Nil)))
→ case xs of
  Nil → z
  Cons(x,xs') →
    op x:x
    y:foldl op:op
      z:z
      xs:xs'
      → case xs of
        Nil → z
        Cons(x,xs') →
          op x:x
          y:foldl op 2
            z
            xs'
            ...
            → / + * x913
            | x
            | * y4
            | y
            2
            → / + * x3672
            | x
            | * y36
            | y
            2
```